ESD-TR-68-440 ESLF MTR-792

MICROPROGRAMMING BRIEFING - - OCTOBER 1968

Thomas L. Connors

MARCH 1969

Prepared for

DIRECTORATE OF PLANNING AND TECHNOLOGY

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

This document has been approved for public release and sale; its distribution is unlimited.

Project 7120
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract F19(628)-68-C-0365

AD0687102

ESD-TR-68-440

MTR-792

MICROPROGRAMMING BRIEFING - - OCTOBER 1968

Thomas L. Connors

MARCH 1969

Prepared for

DIRECTORATE OF PLANNING AND TECHNOLOGY

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 7120

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts

Contract F19(628)-68-C-0365

FOREWORD

This document was produced by The MITRE Corporation, Bedford, Massachusetts in partial fulfillment of Project 7120, Contract F19(628)-68-C-0365.

REVIEW AND APPROVAL

Publication of this technical report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

WILLIAM F. HEISLER, Colonel, USAF
Chief, Command Systems Division
Directorate of Planning and Technology

ABSTRACT

This report presents vugraphs for a briefing on microprogramming fundamentals along with short text explanations.

This page has been left blank to facilitate a "facing-page" format.

Microprogramming was invented in 1951 by Professor M. V. Wilkes of Cambridge University for the purpose of systematizing processor design. It was first implemented on the EDSAC 2 computer at Cambridge.

"... a systematic alternative to the usual somewhat _ad hoc_ procedure used for designing the control system of a digital computer."

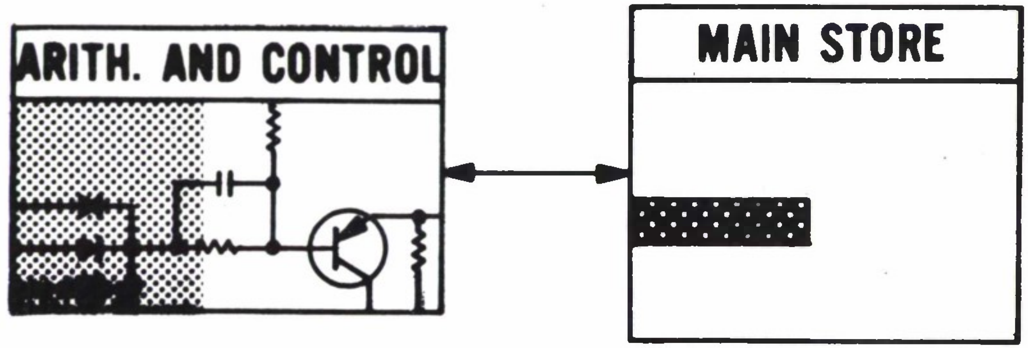
Wilkes 1968

Wilkes observed that the execution of a single computer instruction involves a sequence of steps: for example, fetch from memory, move from one internal register to another, pass through an adder or other logical circuit. In a conventional computer these steps are controlled by _ad hoc_ circuitry. In a microprogrammed computer the steps are identified as micro-instructions and some of the control circuitry is replaced by a control store. Each micro-instruction from the control store calls for the operation of a single step in the interpretation of an ordinary or "native" instruction in the main store.

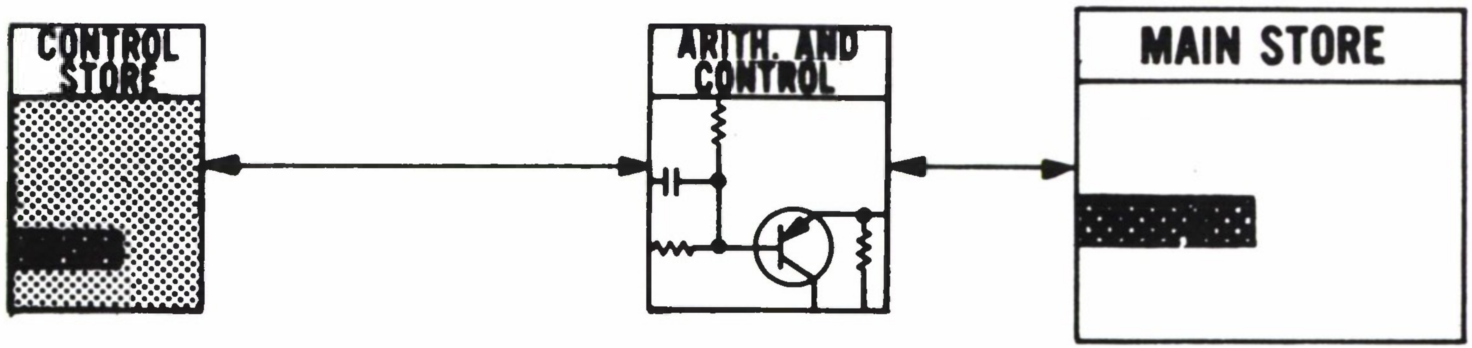
THE MITRE

MICROPROGRAMMING — COMPUTER CONFIGURATIONS

CONVENTIONAL



MICROPROGRAMMED



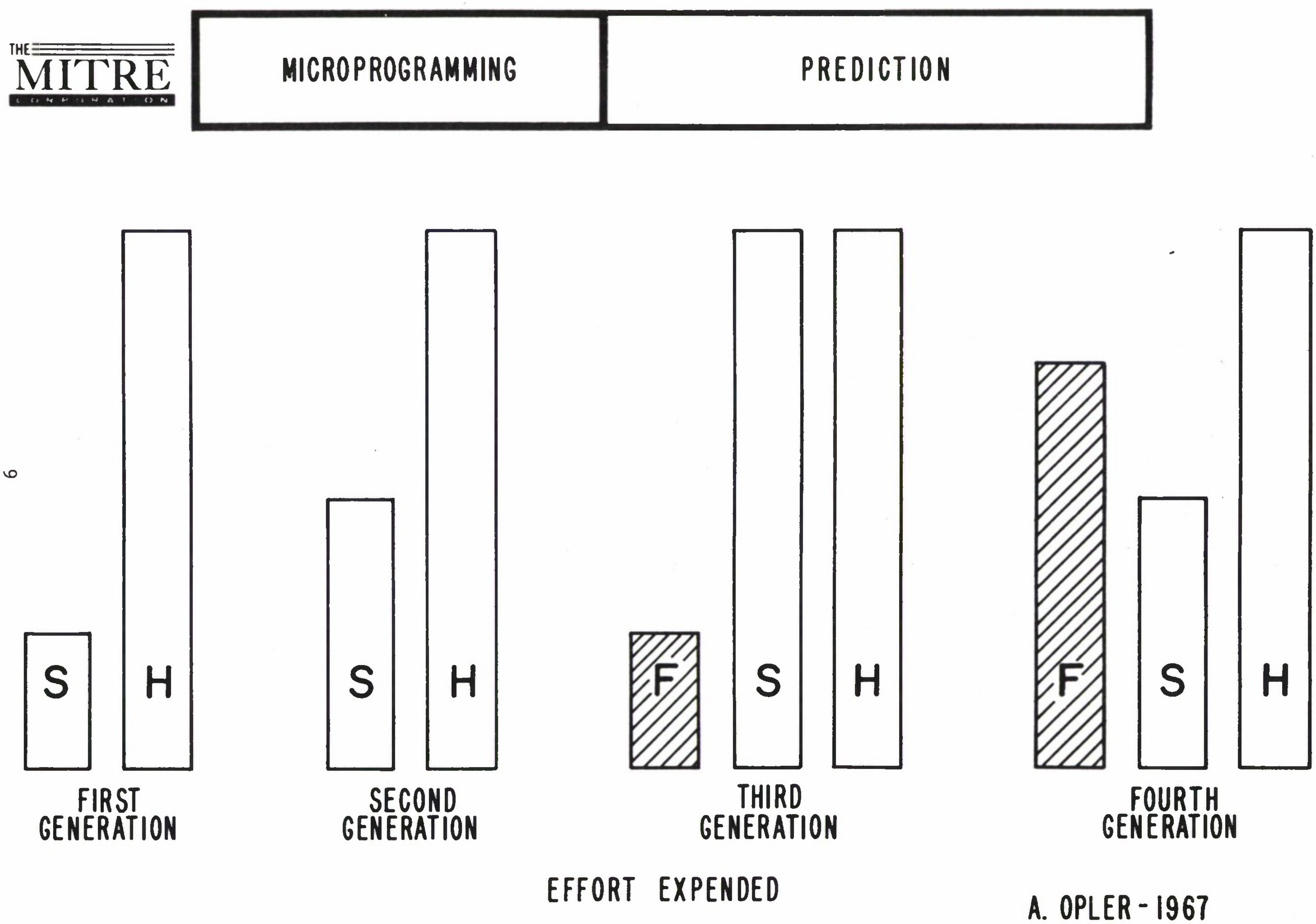
Important elements then are the ideas of subcommands and control stores. Micro-instructions are typically very elementary; sometimes the bits of a micro-instruction correspond to gates in the processor. Control stores, to be discussed later, are typically read-only for speed. More importantly, their contents can be changed or replaced with the effect of implementing a new instruction set in the same hardware.

MICROPROGRAMMING	MICROPROGRAM DEFINITION

- SERIES OF SUBCOMMANDS OR MICRO-INSTRUCTIONS
- KEPT IN CONTROL STORE (USUALLY READ-ONLY)
- CLOSE TO HARDWARE
- CHANGEABLE

Since 1951 the growth of microprogramming was slow until third-generation computers were designed. Most authorities attribute this slow growth to the emphasis on discoveries in software for second and third generation computers. It is well accepted that as much or more effort has been expended on the software of third generation computers as on hardware -- here Ascher Opler predicts that the fourth generation will see more "firmware" or microprogramming than software.

THE MITRE CORPORATION



The number of different computers made by different manufacturers which are microprogrammed is a measure of the influence of microprogramming today. This is only a partial list. The computers range from large to small and from general purpose to the specialized such as the IBM 2841 I/O controller, and the Apollo Guidance computer.

THE MITRE CORPORATION

MICROPROGRAMMING — MICRO-PROGRAMMED COMPUTERS

BURROUGHS	2500
	3500
COLLINS	8401
RCA	SPECTRA 70/35, 70/45
	VIC
HONEYWELL	200/4200
	200/8200
IBM	360/25
	360/30
	360/40
	360/44
	360/50
	360/65
	360/67
IBM (CONT)	360/85
	4PI/CP
	4PI/EP
	2841
INTERDATA	2
	3
	4
STANDARD COMP. CORP	670
	2700
TRW	130
	133
	530
MIT	APOLLO GUIDANCE COMPUTER (DESIGN)

Until recently microprogramming was used primarily by manufacturers who took advantage of the flexibility it allows to make different hardware operate the same instruction sets -- thus developing families of computers such as the IBM 360's and the RCA Spectra 70's. The inverse procedure accomplishes emulation: different contents of the control memory make a single processor perform as two different machines. If the machine architectures vary considerably, extra hardware is required, but emulation has been performed largely through microprogramming. Usually two control memories are attached to the computer at the same time, one containing a microprogram to interpret emulated instructions and the other a microprogram to interpret those of the third generation machine.

Computers have been built with the specific intention of postponing decisions on order set until the application requires orders to be specified -- frequently these multipurpose devices wind up as parts of larger systems. In addition to those shown on the figure, the RCA VIC computer falls in this category. Microprogramming has been applied to fault-isolation and alternative-path operation in high-reliability computers and to the performance of special-purpose counting and sum-of-product operations. An especially extensive set of microprogrammed special operations, and an early "user" application is the set to perform list and character operations in support of the RUSH system. RUSH is operated on an IBM 360/50 by Allen-Babcock Computing. The system is a completely interpretive on-line PL/I for multiple users.

An application of some consequence is the "after the fact" production of a machine by re-microprogramming. IBM transformed the 360/65 into the 360/67 and RCA the S70/45 into the S70/46, albeit with some addition of hardware.

THE MITRE

MICROPROGRAMMING — APPLICATION AREAS

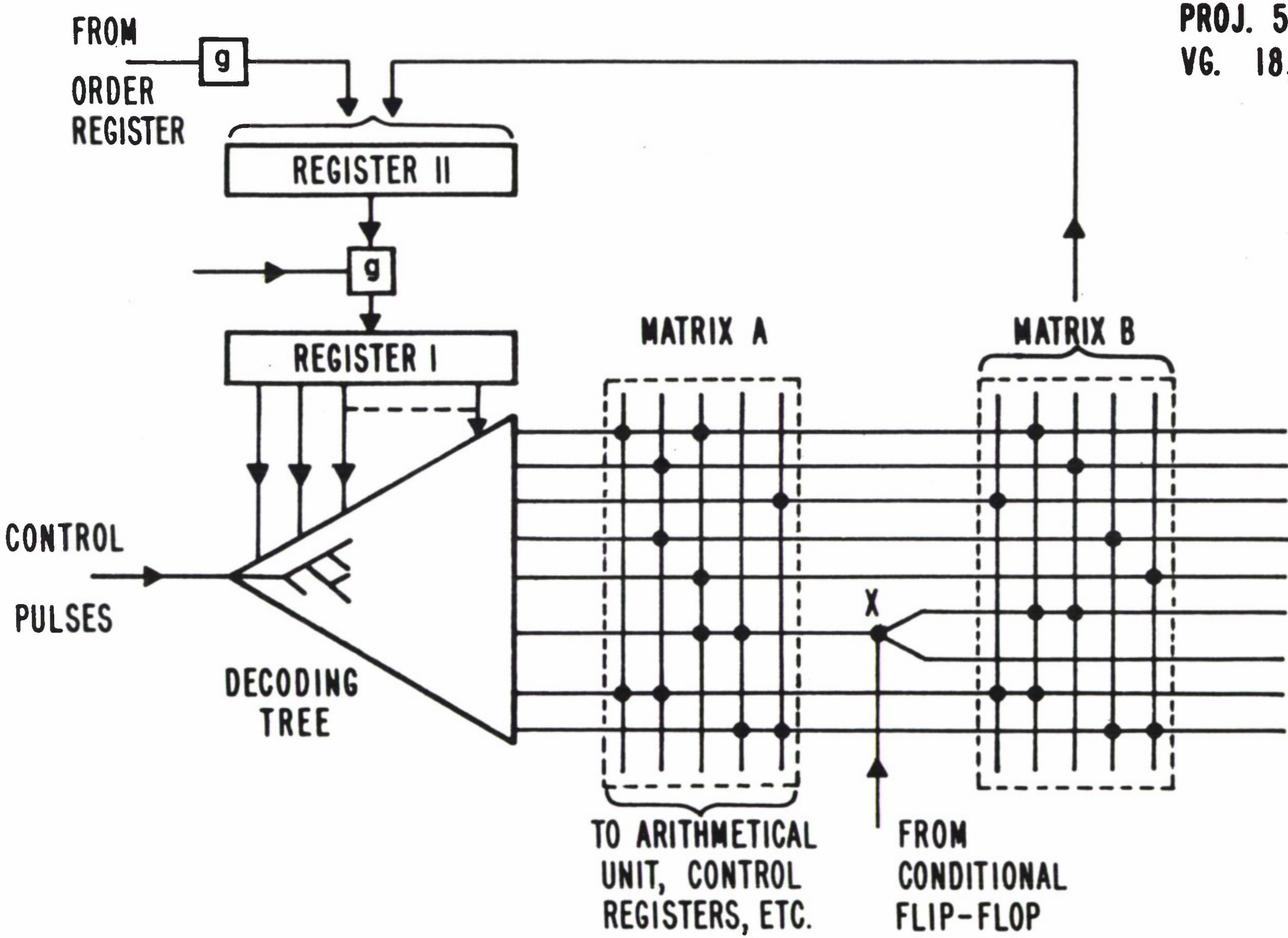
'FAMILIES' OF COMPUTERS	: IBM 360, RCA S70
EMULATION	: IBM 1410 ON IBM 360 IBM 1410, RCA 301 ON S70
MULTIPURPOSE HARDWARE	: IBM 4-PI, IBM AMCS, INTERDATA
RELIABILITY	: APOLLO GUIDANCE COMPUTER DIAGNOSTICS
SPECIAL PURPOSE OPERATIONS	: RUSH SYSTEM, MARK-SENSE MACHINES, RPQ'S
'NEW' COMPUTERS	: IBM 360/67, RCA S70/46

The implementation of a microprogram was originally described by Wilkes with this diagram, taken from his original paper. He said "The microprogram is held in a read-only memory here shown as consisting of two diode matrices, matrix A and matrix B. These are to be regarded as corresponding to two fields in the micro-instruction. The outputs from matrix A are connected to gates in the arithmetic unit and elsewhere in the computer. The access circuits of the memory consist of a decoding tree and the resulting output from matrix A brings about the appropriate micro-operation. The output from matrix B is fed, via a delay circuit, to the address register and thus controls the selection of the next micro-instruction. One of the wires from matrix A is shown as branching before it enters matrix B. The direction taken by the pulse at this branch depends on the setting of the sign flip-flop of the accumulator; the choice of the next micro-instruction to be executed thus depends on whether that number is positive or negative. In a similar way the sequence of control can be made to depend on the setting of other flip-flops in the processor or elsewhere. If desired, the branch can be between the decoding tree and matrix A; in this case the micro-operation itself, as well as the sequence of control, can be made to depend on the setting of the flip-flop in question."

THE MITRE

MICROPROGRAMMING — MICRO-CONTROL UNIT

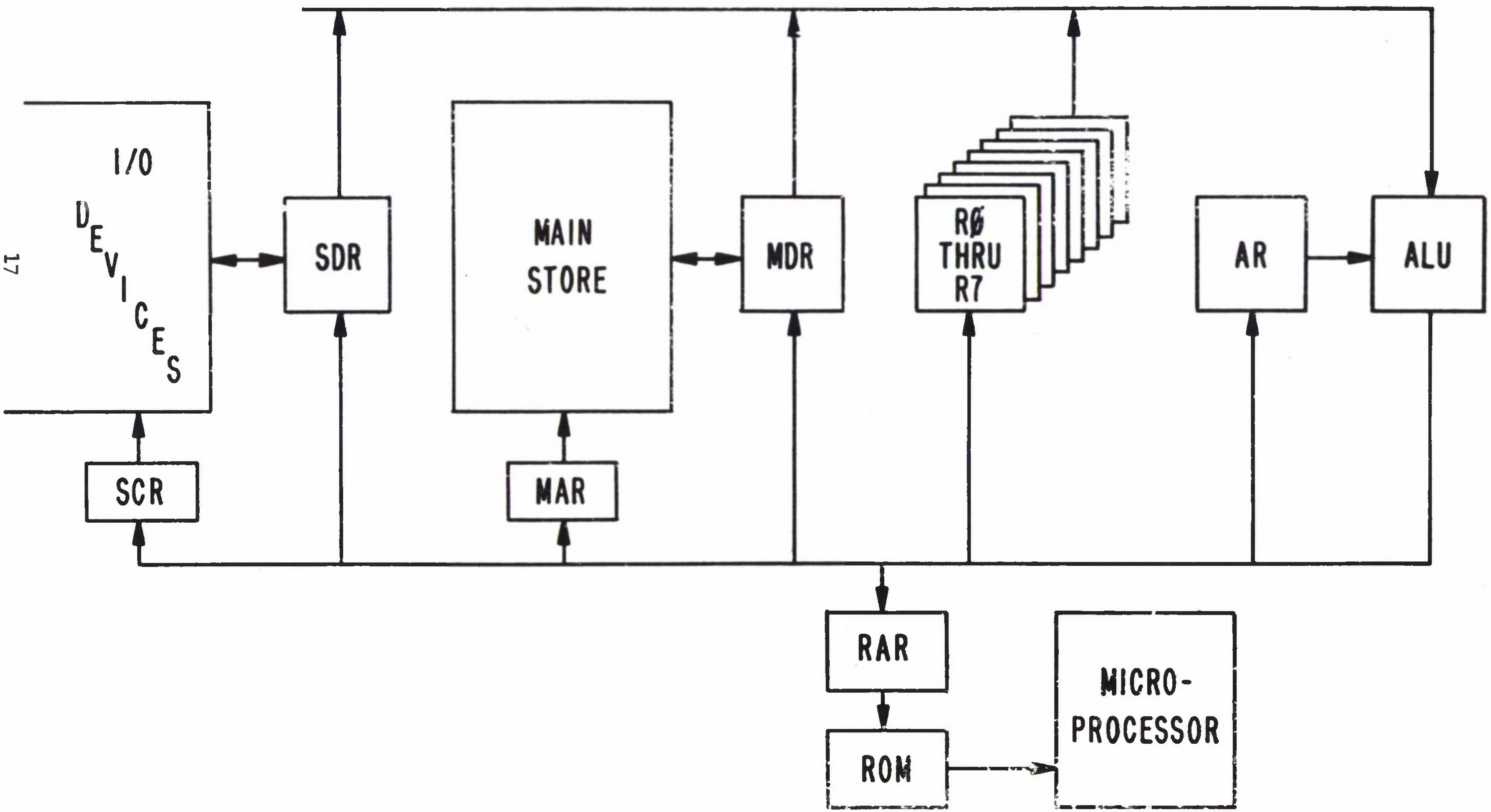
PROJ. 512 V
VG. 18,491



This diagram shows the architecture of the IBM 360/30 basic computer. R, S, T, U, V are 8-bit registers which contain instructions and operands as they are interpreted. Typically an instruction is fetched from the main store into register R, moved through the arithmetic-and-logic-unit to another of the registers, and interpreted from there. Register N is the memory address register for the main store automatically used in fetching operands and instructions. The control store contains the micro-instructions and is addressed by register X. A micro-instruction is fetched into control circuitry which interprets it to open and close the various gates associated with the registers and arithmetic and logical unit. Each of the dark dots on the diagram corresponds to a control point activated by one or more bits of micro-instructions.

THE MITRE CORPORATION

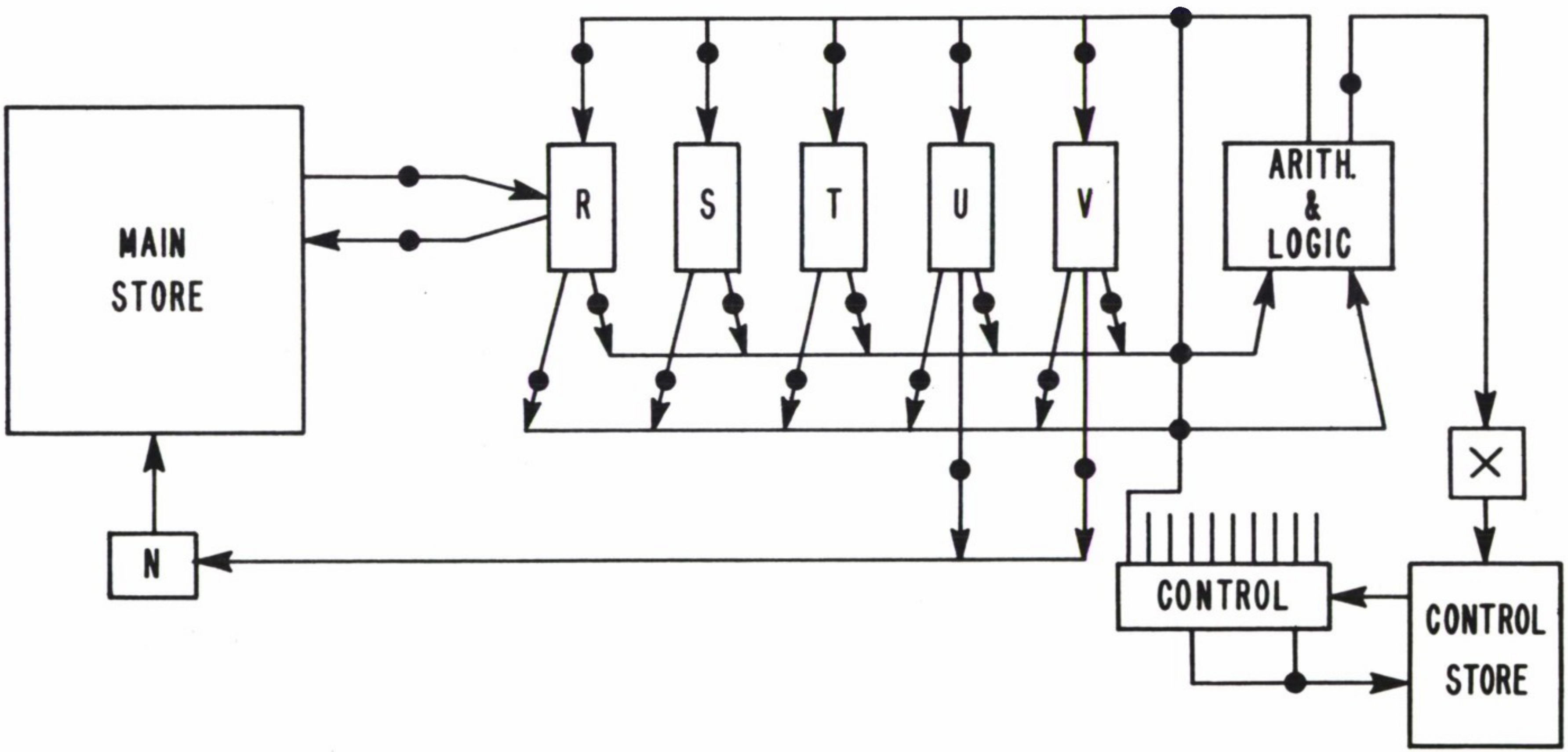
MICROPROGRAMMING | MICRO-MACHINE ARCHITECTURE



This diagram shows the architecture of the Interdata 3 micro-machine and illustrates the similarities between processors organized around busses. The ALU is the arithmetic and logical unit. The micro-processor corresponds to the "Control". The remaining boxes represent 8-bit registers of general utility.

MICROPROGRAMMING

MICRO-MACHINE ARCHITECTURE



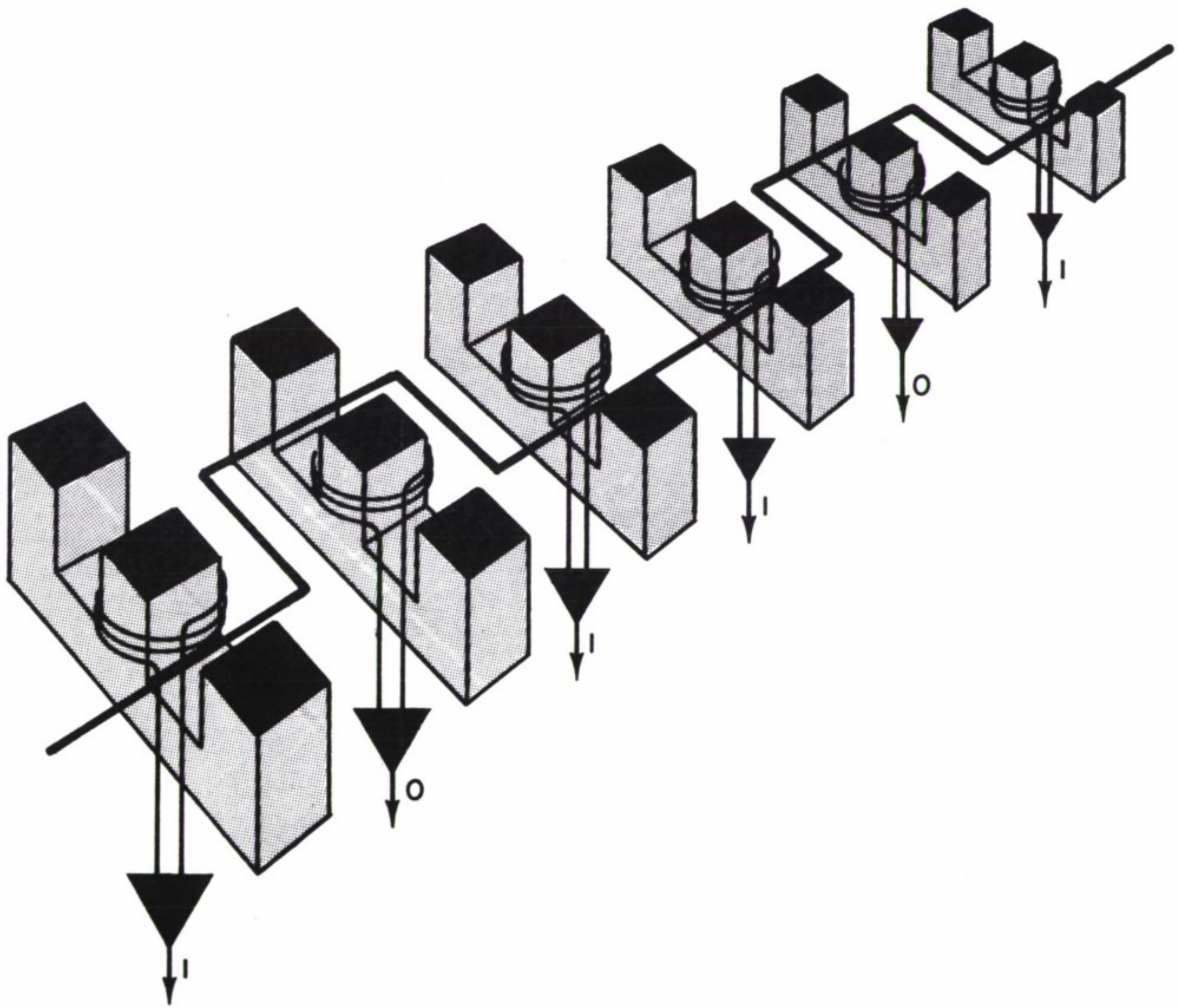
The control store on the third-generation computer is typically a read-only store and read-only store technology has broadened considerably since the diode memories of Wilkes' machine. The IBM 360/40 and RCA Spectra 70 series use transformer read-only stores. The IBM 360/30 and 360/50 use capacitor read-only stores. A developing trend is the use of read/write memories for microprogram storage.

TYPES OF CONTROL MEMORIES

- TRANSFORMER
  - WIRED
  - CONDUCTOR PRINTED ON TAPE
- CAPACITOR
  - TAPE
  - CARD
- READ/WRITE
  - CORE
  - TRANSISTOR

This diagram shows a wired transformer read-only storage similar to that in the RCA Spectra 70/35. The single wire represents a single micro-instruction. Each transformer represents a bit. If the wire is threaded on one side of the transformer secondary, it represents a one, and on the other side a zero. When the wire is pulsed, outputs will be produced only at the sense amplifiers corresponding to one's in the wired micro-instruction. Approximately 500 such wires are laid in one set of transformer cores. Typically, the wires are laid one by one, but braiding techniques which weave "braids" of wires to be laid over transformer cores are now coming into use.

MICROPROGRAMMING | TROS

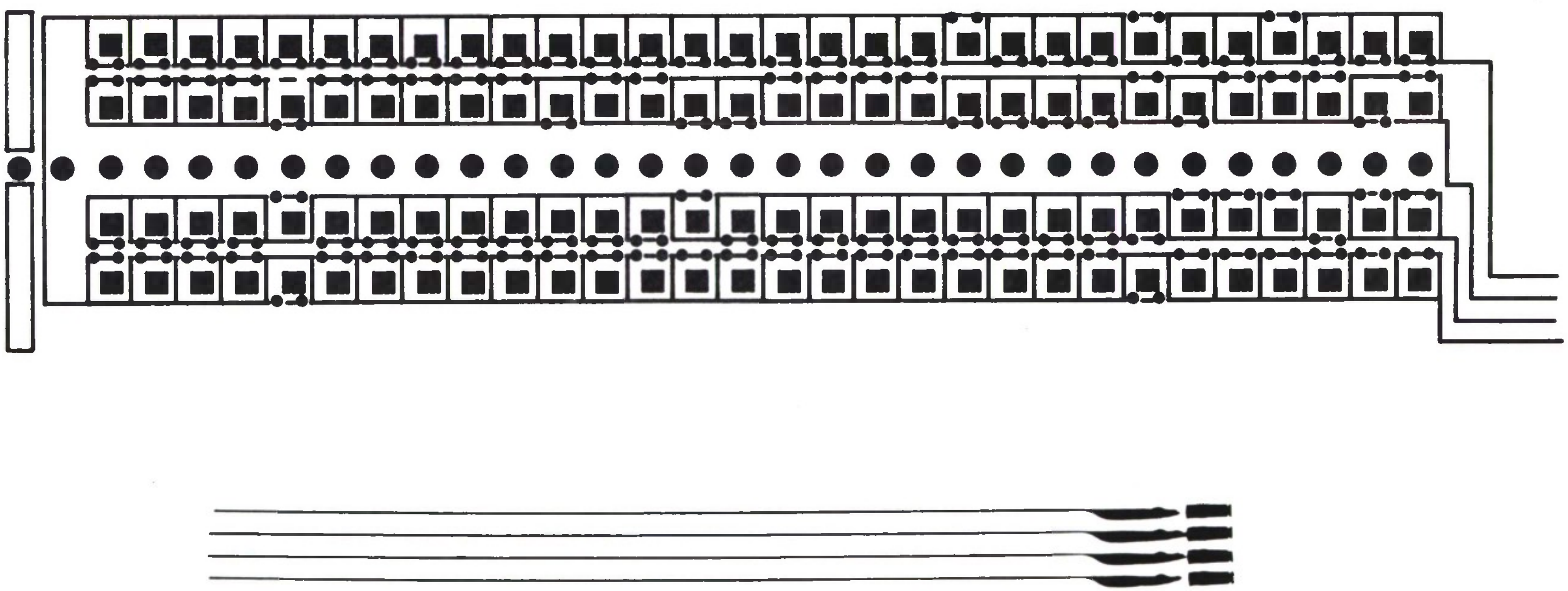


The primary wiring may be printed on a tape and produced automatically as in the 360/40 read-only memory shown here. The microprogram is represented by punches which interrupt lines printed on a mylar tape. Many tapes are laid over many sets of small transformer cores.

THE MITRE CORPORATION

MICROPROGRAMMING

TROS TAPE

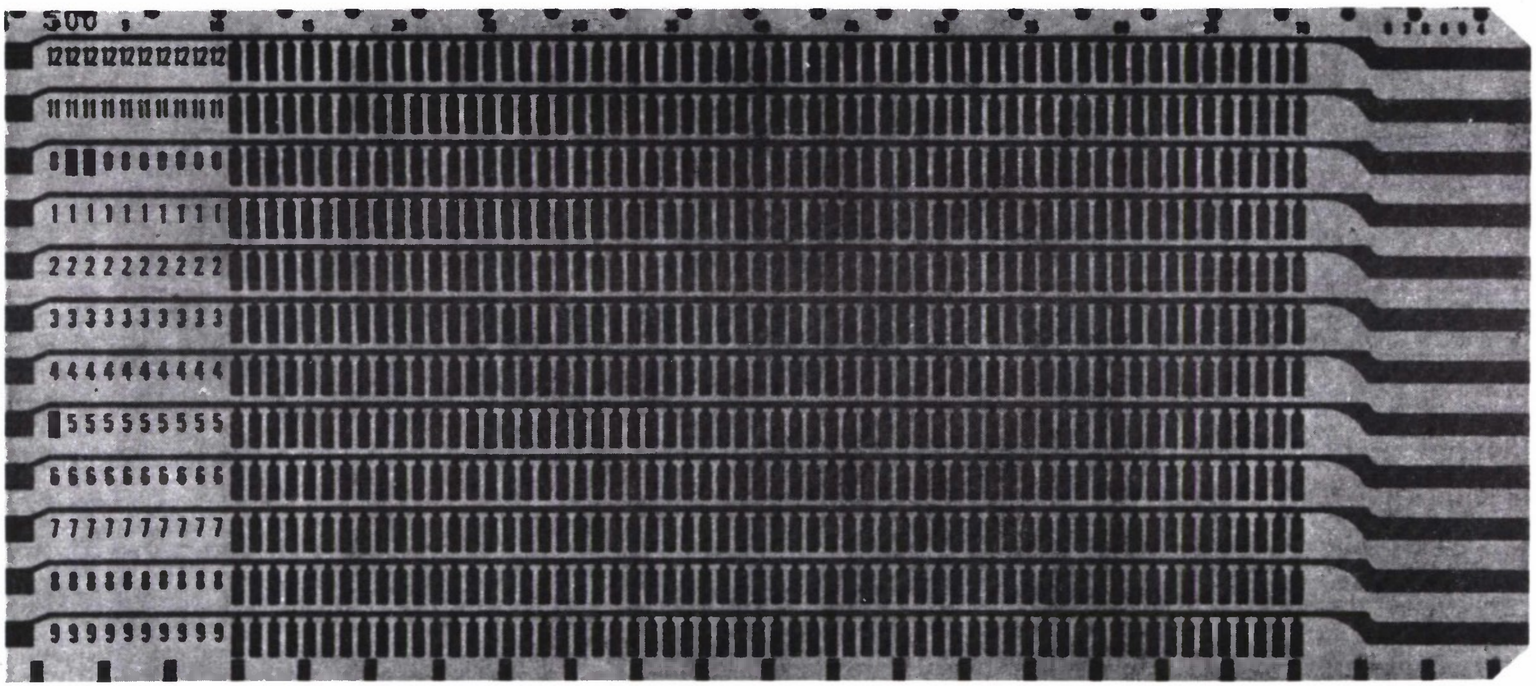


The IBM 360/30 uses a capacitor read-only memory with capacitor plates printed on a special IBM card made of mylar. A microprogram is put on the card by punching out the capacitor plate where no connection is to be made. Each row of the card corresponds to one micro-instruction, i.e., one horizontal wire in Wilkes' diagram. Each column corresponds to a vertical wire connected to a sense amplifier, which is in turn connected to one or more gates in the computer.

THE MITRE CORPORATION

MICROPROGRAMMING

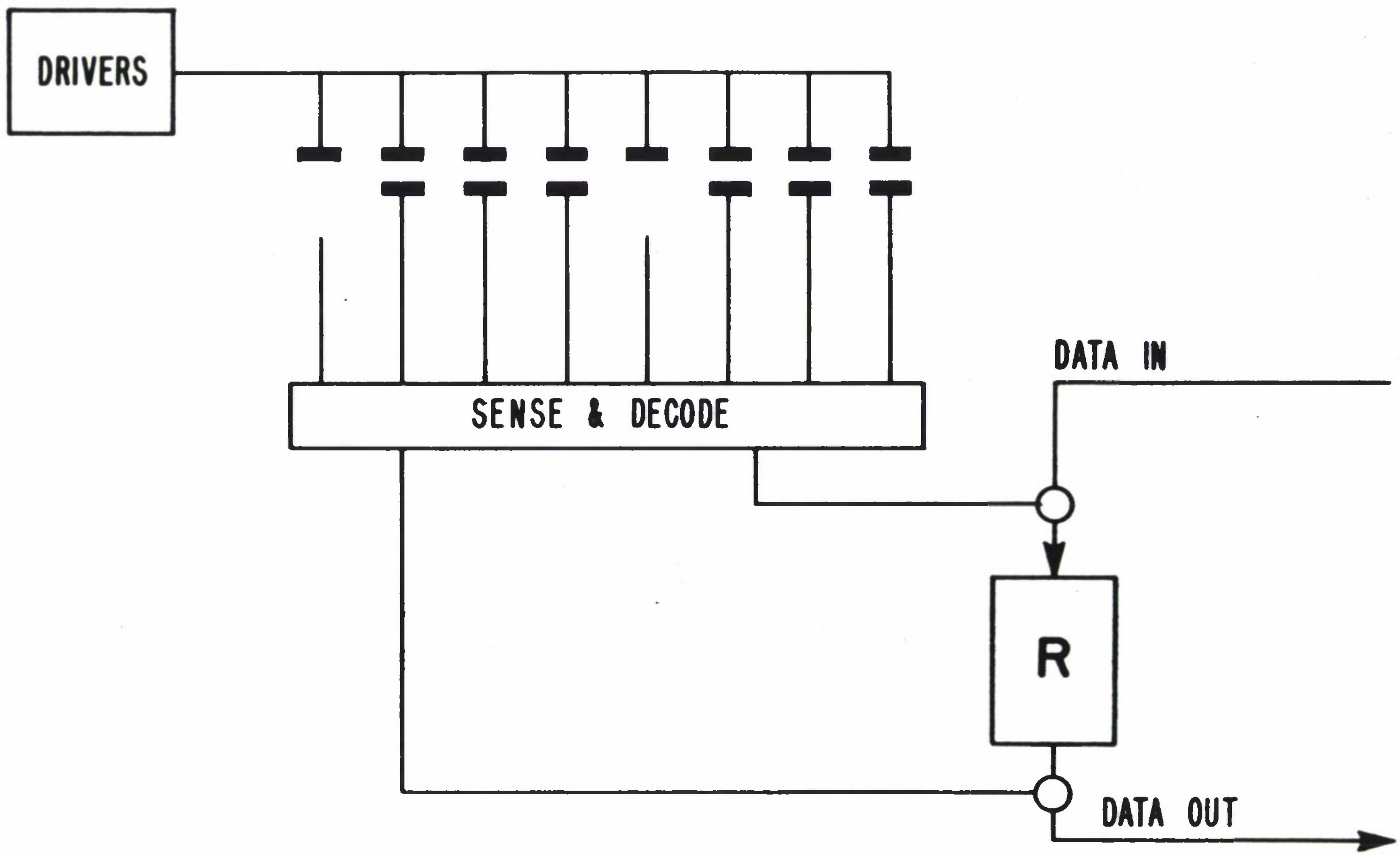
CROS CARD



One row of the card is driven, i.e., one micro-instruction is executed, at a time by a control signal. Outputs from the columns activated might be connected directly to individual gates or, as shown here, connected to a sense and decode network. The decode network accepts n simultaneous inputs and generates 2^n individual outputs which drive gates.

THE MITRE

MICROPROGRAMMING | CROS PRINCIPLES

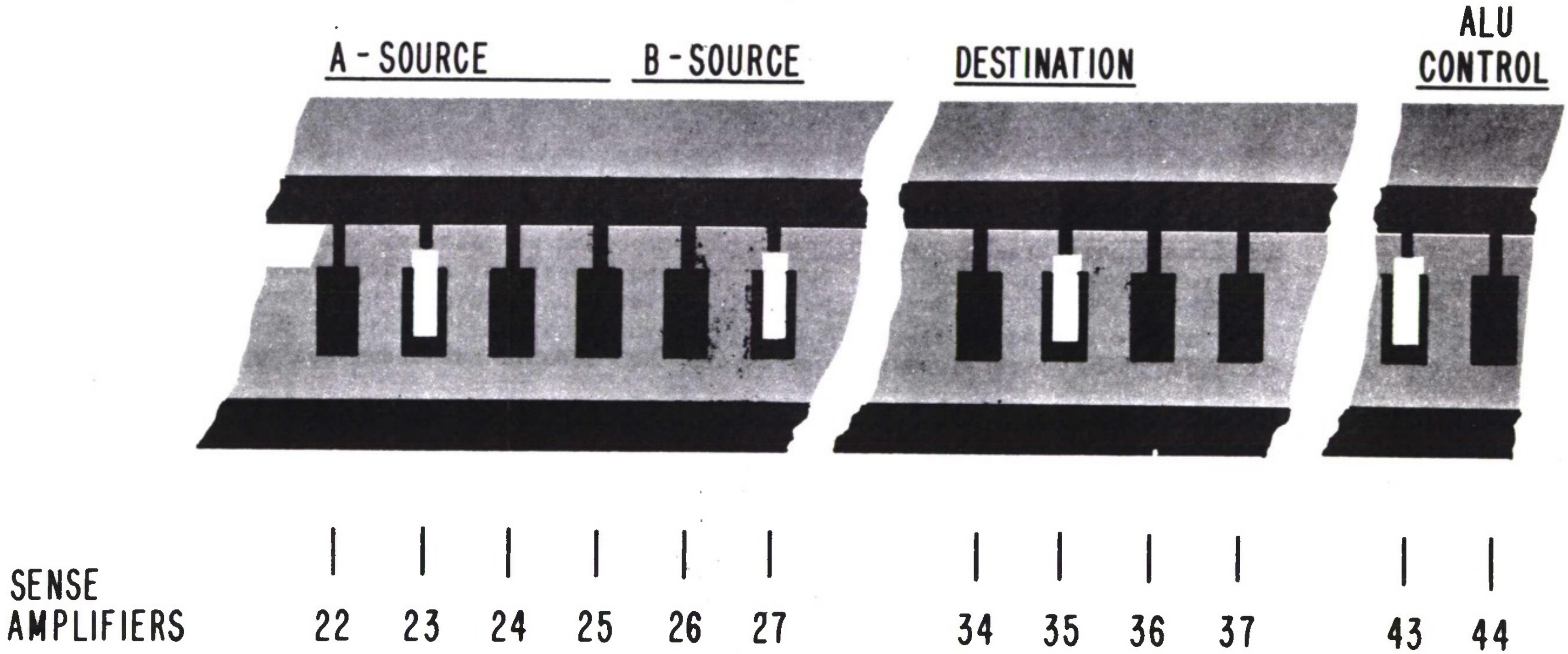


This diagram shows the correspondence of one micro-order "T - D → T" with one row of the capacitor card. The micro-order says "Subtract the contents of register D from those of register T and put the result in register T". It is written on the card in binary: "1011 10 1001 01". The correspondence of columns with sense amplifiers is also shown here.

MICROPROGRAMMING | ROS MICRO-CODE

T — D → T

1011 01 10 1011



We see important microprogram variants on three levels. <u>Architectural variants</u> involve different organizations of the basic operation of the computer, such as direct vs. base-register addressing. <u>Program production aids</u> operate at the interfaces between instruction sequences, and involve such variants as improved procedure-call mechanisms. <u>Specialized operations</u> are the individual instructions done for a single purpose, such as a single instruction square-root or sum-of-products.

THE MITRE

MICROPROGRAMMING — AREAS OF STUDY

HARDWARE	SOFTWARE	FIRMWARE
CPU	—	ARCHITECTURAL VARIANTS
COMPUTER SYSTEM	PROGRAMMING SUPPORT SYSTEM	PROGRAM PRODUCTION AIDS
	GENERALIZED MESSAGE MANAGEMENT DATA MANAGEMENT	SPECIALIZED OPERATIONS
MULT-COMPUTER SYSTEM	APPLICATIONS	

An ESD/MITRE effort in microprogramming is intended to assess the ways in which users of computers can extract from microprogramming benefits so far reserved primarily for manufacturers. The task helped sponsor a seminar attended by about 90 practitioners in the field, acquired a small microprogrammed computer, and is performing microprogramming experiments aimed at studying fundamental architectural variants such as addressing and allocation mechanisms.

THE MITRE CORPORATION

MICROPROGRAMMING	TASK PLANS

- ACM / MITRE SEMINAR - OCT 1968
- CORE SYSTEM FOR INTERDATA
- MICROPROGRAMMING EXPERIMENTS
- POTENTIAL APPLICATION TO
  - PROGRAM PRODUCTION
  - DATA MANAGEMENT
  - MESSAGE MANAGEMENT

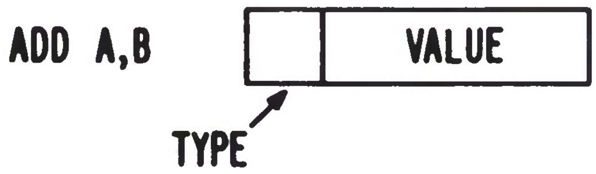
This diagram shows some examples of microprogram application for the purpose of facilitating program production. The first example shows the replacement of three different data forms: binary, decimal, and floating point with a simple self-described data form. The three different add instructions are replaced by a single add instruction. A technique such as this was implemented for the 360/30 by Helmut Weber.

In the second example, a portion of a procedure-fetch algorithm is flow-charted. This algorithm works with a dictionary of procedure locations and makes the procedure fetch independent of the procedure location.

Both algorithms have been coded many times in software. For the second algorithm, we computed comparative timing over a range of computers.

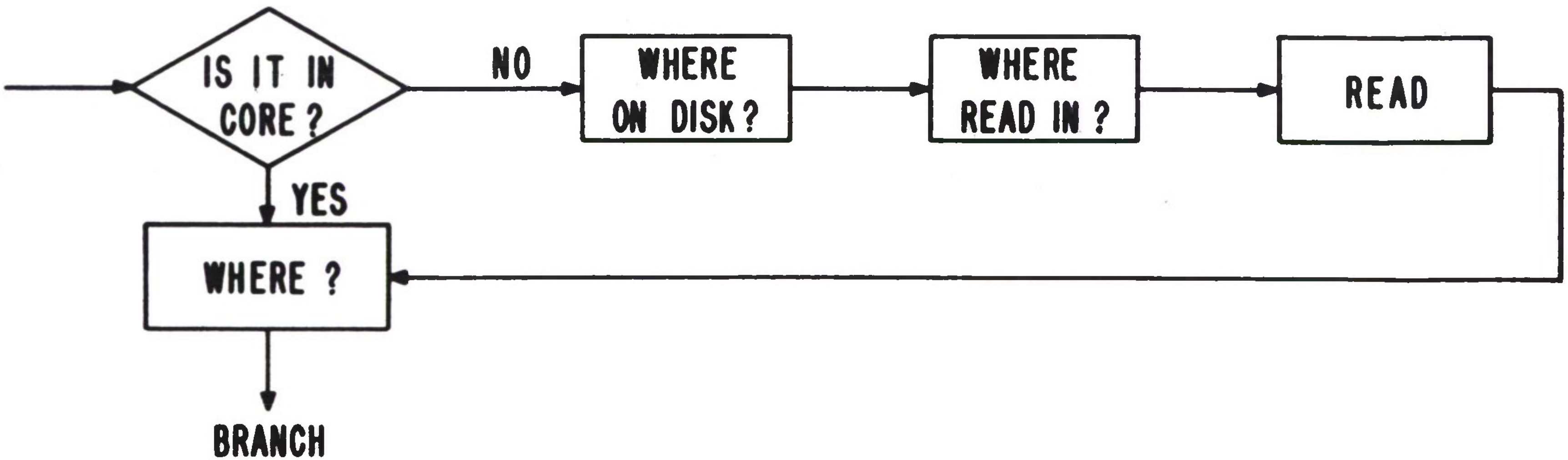
MICROPROGRAMMING | APPLICATION EXAMPLE

DATA FORMS



BADD A, B	0 0 0 0 1 0 1 0
FADD A, B	1 0 0 | 1 0 1 0 0
DADD A, B	0 0 0 1 | 0 0 0 0

PROCEDURE CALLS



The algorithm had originally been programmed for an IBM 1410. It was reprogrammed as an ordinary routine and also microprogrammed for the three other computers shown, and the corresponding run times calculated. The ratio of 71:1 is deceptive if the ratios of memory speeds are not taken into account, but the gain is always more than 6:1 on a single computer. Informal reports of others have yielded gains of between 10:1 to 100:1 for short stretches of code and gains of between 2:1 to 5:1 for large parts of a system.

COMPUTER	TIME FOR SAMPLE PROCESS (MS)		RATIO OF TIMES	
IBM 1410	1.42		71.	
(4.5 μs 1 BYTE)				
INTERDATA 3	.55		28.	
(1.8 μs 2 BYTES 370NS-ROS.)		.034		1.8
IBM 360/30	.39		19.	
(1.5 μs 1 BYTE 750NS-ROS.)		.062		3.
RCA S70/35	.27		13.	
(1.4 μs 2 BYTES 480NS-ROS.)		.019		1.0

In addition to the evident gain in execution speed for the same process, the application of microprogramming offers the potential for more far reaching gains. The flexibility inherent in microprogramming reaches to the fundamental architecture of the machine. Until now, manufacturers have been interested in the application of this flexibility for their own purposes -- presently an opportunity exists for the computer user, who may take steps to realize the efficiencies and economies available in his hardware.

An emulator for a small IBM machine, in essence a new computer in the same hardware, can be rented for approximately $500/month, possibly an example for the user who requires a specialized computer of his own.

THE MITRE CORPORATION

MICROPROGRAMMING	POTENTIAL GAINS

FLEXIBILITY
- ORDER SET
- OPERAND SIZE
- DATA PATHS
- CONTROL PATHS
- TEMPORARY STORAGE
- REDUNDANCY

EFFICIENCY
- ESSENTIAL FUNCTIONS ONLY
- HARDWARE /SOFTWARE MATCH

ECONOMY
- e.g. DIFFERENT COMPUTER FOR $ 500/MONTH, MORE OR LESS

A number of direct application areas for microprogram control are already within the user's grasp, as shown here. However, a new programming technology must be developed to realize the benefits. The microprogrammer is partly logic-designer and must develop a new discipline with new tools and procedures. He is manipulating not only algorithms, but architectures.

THE MITRE CORPORATION

MICROPROGRAMMING	POTENTIAL APPLICATION AREAS

DATA MANAGEMENT	— MORE EFFICIENT ALLOCATION & ADDRESSING — INSTRUMENTATION OF PROCEDURE & DATA REFERENCES — MORE SIMPLIFICATION FOR USER — BETTER PERFORMANCE
COMMUNICATIONS	— MORE EFFICIENT CODE CONVERSION — MESSAGE ROUTING — HEADER PROCESSING
TIME SHARING	— USER SEPARATION — PRIORITY — LOW OVERHEAD SWITCHING — SECURITY
RELIABILITY	— REDUNDANT CODE

The biggest news in microprogramming is the adoption by manufacturers of the changeable control store, which implies greater availability to the user of micro-machines. The IBM 360/25 and 360/85 are examples, and the SCC 6700 is specifically sold for the purpose.

The availability of LSI memories large enough to contain the micro-coded description of a machine will imply a greater need for regularity of control circuits -- we may expect that regularity to be achieved by microprogrammed design.

The development of inexpensive small computers makes the "special-purpose" general-purpose computer attractive. Washington State University has a small microprogrammed teletype concentrator in operation; several organizations expect to microprogram computers for the direct execution of algorithmic languages such as FORTRAN and PL/I.

Microprogramming itself has not taken advantage of the benefits of the software art -- computers, debugging tools and production systems for firmware are in their infancy.

It is in the application of microprogramming to the area between software and hardware that the greatest gains are to be expected -- architectures of coming computers are likely to incorporate user-defined operating system components in place of some of the large software supervisors presently in use. Likewise the architecture of the machine will be made more appropriate to the user's software requirements.

TRENDS

- CONTROL STORES
  - Changeable
  - LSI

- NEW "COMPUTERS"
  e.g. - Communications
  - Higher-Level-Language Computers

- SOFTWARE PRINCIPLES APPLIED TO MICRO-CODE

- MICRO-CODE TO FACILITATE SOFTWARE PRODUCTION
  - Operating Systems
  - Software Environments

10/10/68

THE MITRE

MICROPROGRAMMING	INTENTION

"... a systematic alternative to the usual somewhat ad hoc procedure used for designing the control system of a digital computer."

Wilkes 1968

10/10/68

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* The MITRE Corporation, Bedford, Massachusetts	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

MICROPROGRAMMING BRIEFING - - OCTOBER 1968

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
N/A

5. AUTHOR(S) *(First name, middle initial, last name)*

Thomas L. Connors

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
MARCH 1969	47	

8a. CONTRACT OR GRANT NO. F19(628)-68-C-0365 b. PROJECT NO. 7120 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ESD-TR-68-440 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* MTR-792

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
N/A	Directorate of Planning and Technology, Electronic Systems Division, Air Force Systems Command, USAF L. G. Hanscom Field, Bedford, Massachusetts

13. ABSTRACT

This report presents vugraphs for a briefing on microprogramming fundamentals along with short text explanations.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
COMPUTERS						
MICROPROGRAMMING						